Exact volume preserving skinning with shape control

Damien Rohmer[1,2], Stefanie Hahmann[1] and Marie-Paule Cani[1,2]

[1]LJK Lab, Grenoble Universités, France
[2]INRIA, France

Abstract

In the real world, most objects do not loose volume when they deform: they may for instance compensate a local compression by inflating in the orthogonal direction, or, in the case of a character, preserve volume through specific bulges and folds. This paper presents a novel extension to smooth skinning, which not only offers an exact control of the object volume, but also enables the user to specify the shape of volume-preserving deformations through intuitive 1D profile curves. The method, a geometric post-processing to standard smooth skinning, perfectly fits into the usual production pipeline. It can be used whatever the desired locality of volume correction and does not bring any constraint on the original mesh. Several behaviors mimicking the way rubber-like materials and organic shapes respectively deform can be modeled. An improved algorithm for robustly computing skinning weights is also provided, making the method directly usable on complex characters, even for non-experts.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Animating characters is a complex process, which needs to fit into the standard production pipe-line for efficient use by artists. Skeleton based deformation is widely used to model articulated motion, the most popular method being *Skeleton Subspace Deformation* (SSD), based on weighted frames blending and also called *Smooth Skinning*. Despite of the intuitive control it brings, large deformations lead to undesirable effects such as very visible loss of volume near articulations. Results therefore lack realism and fail to mimic natural behavior, such as, in the case of characters, muscles inflating in the transverse direction when they contract or the flesh and skin forming large folds when a character bends.

Insuring a natural-looking behavior through automatic local volume preservation should ideally be done without requiring any modification of the standard animation inputs. Moreover, to enable fast and intuitive settings, handles on this preservation should be easy to parameterize, even on coarse meshes.

This article presents a fully geometric framework to exactly control the volume of skinned characters throughout an animation. The method is based on a post-processed volume correction, independently applied at each frame, so it can be plugged into preexisting deformation processes. The control tools we provide enable to locally imitate various materi-

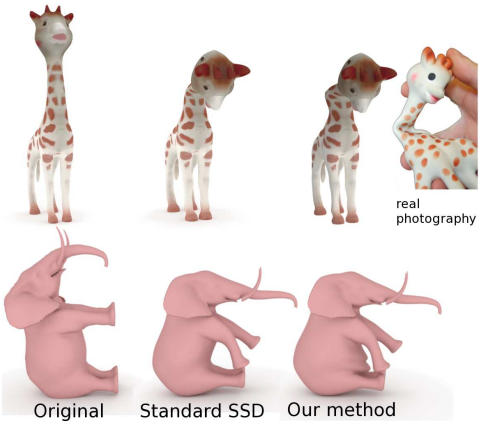

Figure 1: *Top: Rubber giraffe toy where anisotropic inflation is used to compensate the loss of volume and mimics a rubber tube full of air. Bottom: Making an elephant sit where the original volume is integrally restored thanks to belly bulges, intuitively controlled through 1D profile curves.*

als, from rubber to organic behaviors. They include 1D profile curves to quickly sketch the shape the volume correction should exhibit.

1.1. Previous Work

Multiple approaches have been designed to limit the well known artifacts of the SSD method such as the loss of volume occurring for large deformations [LCF00]. Training example based approaches achieve realistic deformations with real-time performances. Complex interpolation algorithms [LCF00, SRC01, WP02], and methods based on the addition of extra frames [MG03], direct mesh deformation [KJP02], or the fitting of regression parameters [WPP07] were studied, often requiring the addition of extra static key poses.

Farther from the standard animation pipe-line, exact volume preserving deformations were developed in order to represent muscular models [HJCW06], and alternative mesh representation were used, resulting in large matrix system to be solved [HSL*06].

In this paper, we propose a method that only requires the input of a standard animation production, namely a single surface mesh in a resting pose.

Non linear interpolation of frame orientations [Ale02, KZ05] improves the skinning method itself and can be computed at interactive rates. In particular, the use of dual quaternion [KCZO08] solves the torsion artifacts and reduces the loss of volume. However, the final aspect of the deformation is even more sensitive to the choice of the skinning weights, which are hard to set, and cannot be used to parameterize complex effects such as muscles bulging or skin folding. Since it acts as a post-process, our method can be used in conjunction to these interpolation mechanisms, to bring an improved control on the final shape.

Closer to our approach, Angelidis and Singh [AS07] developed a volume preserving skinning algorithm, based on a powerful embedding into volumetric space. A degree of freedom is left to the artist to control the final shape, although its combination with skinning weights variation along the mesh makes this control somewhat indirect. Contrary to our approach, solving the problem requires some numerical integration, which makes the method less efficient on large meshes and does not fit into the classical pipeline. Moreover, deformation is directly derived from the expression of SSD, so the method would not be easy to extend to better interpolation schemes such as dual quaternions [KCZO08]. An exact geometrical post-process correction to volume variations, which can be, like ours, combined to any skinning scheme, was also investigated in [FTS08]. The great benefits of this approach is its direct control by painting the exact desired amount of correction over the surface. However, this does not allow a minimal per-vertex displacement to compensate the volume error.

In our previous work [RHC08], we already presented an automatical volume correction method. While the objective minimization equations are closely related, the current method improves some key features:

- An exact local volume preservation correction is ensured for any deformation. The previous work only solved a linearized formulation of the constraint and approximate the local volume change.
- The final shape can be intuitively controled using 1D-profil curves while the previous approach almost entirely relied on skinning weights values which are hard to define.
- The deformation space spans $\mathbb{R}^{3N}$ with N being the number of vertices. The volume-correction in the previous work was only acting along a predefined N-dimensional space, reducing the range of possible shapes.
- Quadrilateral meshes are treated using bilinear patches instead of requiring triangulation leading to visual artifacts.
- The new method enables a volume correction deformation from multiple bones to overlap. In [RHC08] bones influences were restricted to segmented regions given by the main skinning weight's value. This segmented definition did not enable regions with many bones' dependencies to be easily deformed.

1.2. Contributions

This paper proposes a simple and easy to use framework, consistent with standard modeling tools, for exactly preserving (or controlling) the volume of a skinned character. The method can handle large deformations. It offers some intuitive shape control by applying the correction according to 1D profiles, which can be arbitrarily defined, independently from the input mesh. The expected visual appearance is achieved without any loss of volume and runs at interactive rates.

Section 2 presents the theoretical basis of our approach, namely a per-vertex displacement field, derived for both triangular and quadrangular meshes, which exactly preserves the volume after any deformation. Section 3 introduces our approach for offering the appropriate local control of the resulting deformation. First, local frames defining the main orientations of volume preserving deformations are set up. In such frames, 1D profile curves are defined to shape the locality and distribution of volume corrections. Next, an iterative method to handle multiple articulations simultaneously is presented. We give examples of various natural effects easily obtained with our method. Section 4 details the application of this method to animated characters: we first present an improved automatic skinning weight computation, which robustly handles animation of complex characters such as animals. Then user-control is highlighted by presenting the effects we can obtain when characters articulate. The last section discusses our approach and gives directions for future work.

2. Exact volume compensation

This section presents our general solution to the exact preservation of volume during deformation. We first extend the trilinear expression of the enclosed volume of a triangular mesh to the case of quadrangular meshes often used in character animation. We next derive an exact closed form solution to the problem of minimizing volume variations when the mesh deforms.

2.1. Expressing the enclosed volume of the mesh

Let S be a closed manifold triangular surface. Its vertex positions are denoted by $\mathbf{p}_i = (x_i, y_i, z_i)$, $i = 0, \ldots, N-1$. The signed volume V of S is then given by

$$V = \sum_{\text{face}(l,m,n) \in S} \frac{z_l + z_m + z_n}{6} \begin{vmatrix} (x_m - x_l) & (y_m - y_l) \\ (x_n - x_l) & (y_n - y_l) \end{vmatrix}, \quad (1)$$

where $\text{face}(l,m,n) = \Delta(\mathbf{p}_l, \mathbf{p}_m, \mathbf{p}_n)$ is a triangle of S. This trilinear expression can be derived from the volume functional for smooth surface, see [HML99, GOMP98, DMSB99]. It corresponds to the sum of the (signed) volumes of the prisms spanned by the surface triangles and their projections onto the xy-plane (see fig. 2).

Let us denote $\mathbf{p_x} = (x_0, \ldots, x_{N-1}) \in \mathbb{R}^N$ the vector of the x-coordinates of the N vertices, and $\mathbf{p} = (\mathbf{p_x}, \mathbf{p_y}, \mathbf{p_z}) \in \mathbb{R}^{3N}$. $\mathbf{p_y}$ and $\mathbf{p_z}$ are defined analogously. Using the trilinearity of V, the volume can also be written in terms of a scalar product

$$V(\mathbf{p}) = < \mathbf{p}_x, \nabla_{\mathbf{x}} V > = < \mathbf{p}_y, \nabla_{\mathbf{y}} V > = < \mathbf{p}_z, \nabla_{\mathbf{z}} V >,$$

where the gradient vectors $\nabla_{\mathbf{x}} V, \nabla_{\mathbf{y}} V, \nabla_{\mathbf{z}} V \in \mathbb{R}^N$ can be analytically expressed by differentiating the volume expression (1).

Artists usually work with quad-dominant meshes when dealing with animated characters to define easily lines and directions of interest during deformation. Using the classical volume expression on such a mesh requires a triangulation. This triangulation can introduce unwanted artifacts in the volume gradient due to the artificial change in the connectivity as ∇V depends on the one ring of vertices. We suggest to treat quadrilateral faces directly by computing the sub-volume associated to the bi-linear patch interpolating the four vertices $(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$. In this case, the exact analytical signed volume can be expressed as

$$V_q = \mathbf{z} \mathrm{M} \mathbf{k}^T \quad \text{with} \quad \mathrm{M} = \frac{1}{36} \begin{pmatrix} 4 & 2 & 2 & 1 \\ 2 & 4 & 1 & 2 \\ 2 & 1 & 4 & 2 \\ 1 & 2 & 2 & 4 \end{pmatrix}. \quad (2)$$

$\mathbf{z} = (z_0, z_1, z_2, z_3)$, and $\mathbf{k} = (k_0, k_1, k_2, k_3)$ being the vector whose entries are the determinants:

$$\begin{cases} k_0 = (x_1 - x_0)(y_2 - y_0) - (y_1 - y_0)(x_2 - x_0) \\ k_1 = (x_1 - x_0)(y_3 - y_1) - (y_1 - y_0)(x_3 - x_1) \\ k_2 = (x_3 - x_2)(y_2 - y_0) - (y_3 - y_2)(x_2 - x_0) \\ k_3 = (x_3 - x_2)(y_3 - y_1) - (y_3 - y_2)(x_3 - x_1) \end{cases}.$$

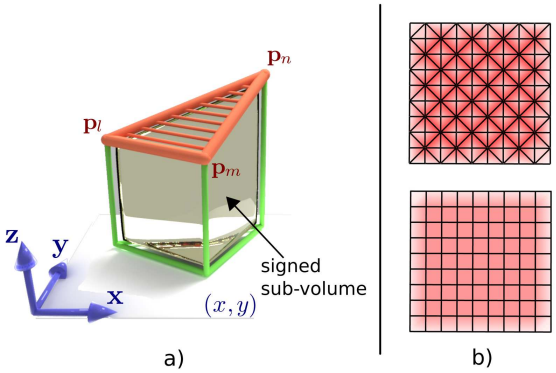

Figure 2: *a) Prism obtained by projecting the triangle $(\mathbf{p}_l, \mathbf{p}_m, \mathbf{p}_n)$ onto the (x,y) plane. The summation of all the signed sub-volumes give the interior volume of the triangular mesh. b) Color encoding of $\|\nabla V\|$ over a triangulated quad-mesh (top) using differentiation of eq. 1, and using eq. 2 directly on the bilinear patches (down). Note the periodical pattern appearing in the first case due to the change of valence.*

Denoting V_t and V_q the sub-volume associated respectively to the triangle t and the quad q, the volume of a complete quad-dominant mesh made of N_t triangles and N_q quads is given by

$$V = \sum_{t=0}^{N_t - 1} V_t + \sum_{q=0}^{N_q - 1} V_q \, .$$

It follows that for any mesh made of quadrilateral and/or triangular faces, the volume expression is still a trilinear function and thus its gradient ∇V can be expressed analytically.

2.2. An exact closed-form correction in three steps

Two meshes are now considered. An original surface $\overline{S}$ whose vertices are denoted by $\overline{\mathbf{p}}$, and a pointwise deformed surface S with vertices $\mathbf{p}$. The global change of volume between these two meshes is defined by $\Delta V = V(\mathbf{p}) - V(\overline{\mathbf{p}})$.

The goal is to compute a correction vector $\mathbf{u} = (\mathbf{u_x}, \mathbf{u_y}, \mathbf{u_z}) \in \mathbb{R}^{3N}$ such that the volume of the corrected mesh $V(\mathbf{p} + \mathbf{u})$ equals the original one and such that the correction is as small as possible.

This correction vector is found as a solution of a minimization problem:

$$\begin{cases} \min & \|\mathbf{u}\|^2_{\mathbb{R}^{3N}} \\ \text{constraint to} & V(\mathbf{p} + \mathbf{u}) = V(\overline{\mathbf{p}}). \end{cases} \quad (3)$$

This constrained minimization problem can be converted into an unconstrained problem using the technique of Lagrange multipliers: $\max_\lambda \min_{\mathbf{u}} \Lambda(\mathbf{u}, \lambda)$, where

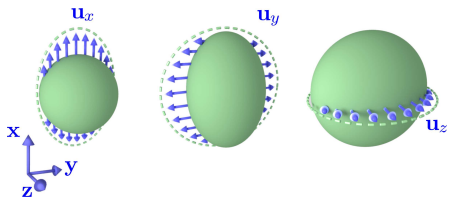

Figure 3: *An exact three step correction method. Firstly, the correction is only applied along x direction. The second step corrects along the y direction, and the third one along the z direction.*

$\Lambda(\mathbf{u},\lambda) = \|\mathbf{u}\|^2 + \lambda(V(\mathbf{p}+\mathbf{u}) - V(\overline{\mathbf{p}}))$. The vector $\mathbf{u}$ giving the correct volume satisfies $\nabla\Lambda(\mathbf{u},\lambda) = 0$. Note that the constraint is not linear. Our approach to solve the problem is inspired by a technique presented in [Elb00] for Bézier curves and surface. We extend this work to polygonal surfaces and derive an exact analytical solution using the trilinearity of the expression of the volume. Furthermore we add the possibility to choose a privileged direction to the volume correction field $\mathbf{u}$ solution of eq. (3). Our solution is as follows (see fig. 3):

1. The correction $\mathbf{u}$ is first only considered in the x direction and the mesh is deformed such that the volume is corrected by a given percent denoted μ_0.
2. In a second step, the new corrected mesh undergoes an other deformation applied only in the y direction. In this case, the volume is corrected by μ_1 percent.
3. In a third step, $\mathbf{u}$ is considered along the z direction and the volume is corrected by μ_2 percent.

If $\mu_0 + \mu_1 + \mu_2 = 1$, the volume of the final mesh equals the volume of $\overline{S}$. The method works whatever the definition of x, y and z. Later, we will use a local definition of such frames.

To derive the closed form solution of the first correction step, the solution to the constrained problem must be a root of the gradient of the Lagrangian Λ_1, with

$$\Lambda_1(\mathbf{u}_x,\lambda) = \|\mathbf{u}_x\|^2 - \lambda\Big[V(\mathbf{p}_x+\mathbf{u}_x,\mathbf{p}_y,\mathbf{p}_z) - V(\overline{\mathbf{p}}) + (1-\mu_0)\Delta V\Big] \ .$$

One can note that

$$V(\mathbf{p}_x+\mathbf{u}_x,\mathbf{p}_y,\mathbf{p}_z) - V(\overline{\mathbf{p}}) = <\mathbf{u}_x, \nabla_{\mathbf{x}} V> - \Delta V \ .$$

Therefore, the solution of equation $\nabla\Lambda_1 = 0$ is analyticaly expressed as $\mathbf{u}_x = \mu_0 \Delta V \frac{\nabla_{\mathbf{x}} V}{\|\nabla_{\mathbf{x}} V\|^2}$.

Let $S^\star$ be the surface obtained after the correction in the x direction. In the second step, $S^\star$ is deformed along the y direction such that $V(\mathbf{p}_x+\mathbf{u}_x,\mathbf{p}_y+\mathbf{u}_y,\mathbf{u}_z) = V(\overline{\mathbf{p}}) - (1-\mu_0-\mu_1)\Delta V$. Following the same method, the solution for this step is given by $\mathbf{u}_y = \mu_1 \Delta V \frac{\nabla_{\mathbf{y}} V^\star}{\|\nabla_{\mathbf{y}} V^\star\|^2}$, where $V^\star$ represents the volume associated to $S^\star$.

Similarly, let us denote $S^{\star\star}$ the surface deformed in x and y. The final z correction is then given by $\mathbf{u}_z = \mu_2 \Delta V \frac{\nabla_{\mathbf{z}} V^{\star\star}}{\|\nabla_{\mathbf{z}} V^{\star\star}\|^2}$.

At the end of these three steps, the initial volume is exactly restored. The computation cost of this exact correction is roughly the same as three evaluations of the volume, so it does not affect the efficiency too much. The coordinate decoupling introduces a dependency on the order of the operation which will be discussed in 5.

Note that the same method could be used to animate changes of volume rather than to keep it to a constant value.

3. Local control of the deformation

In the method we have just presented, volume was computed and corrected in a global way. In character animation, however, the preservation of volume should rather acts locally to better model the behavior of soft tissues. Moreover various materials should exhibit different deformation shapes. Anisotropic effects are for instance desirable for modeling muscles acting in specific directions. This section presents our method for achieving exact volume preservation locally while controlling the shape of the correction.

We first present a method for building meaningful local frames for skeleton based deformation, in which some specific weights control the effects of volume corrections. They can be easily specified by the user through 1D profile curves. We then present an iterative algorithm to handle multiple articulations simultaneously.

3.1. Localizing the deformation

To prevent global inflation caused by the volume correction step, we distribute the deformation with respect to a set of weights $\gamma = (\gamma_1,\ldots,\gamma_N)$, similarly to the approaches in [HJCW06, RHC08, FTS08]. In our case, this local weighting is achieved by re-defining the norm used in eq. (3) by $\|\mathbf{a}\|^2_\gamma := \sum_{k=0}^{N-1} \frac{a_k^2}{\gamma_k}$. In this case, the weighted closed-form solution for the displacement vector $\mathbf{u}_k$ at vertex k is given by

$$\mathbf{u}_k = \gamma_k \Delta V \left(\mu_0 \frac{\nabla_{\mathbf{x}} V(\mathbf{p}_k)}{\|\nabla_{\mathbf{x}} V\|^2_{1/\gamma}}, \mu_1 \frac{\nabla_{\mathbf{y}} V^\star(\mathbf{p}_k)}{\|\nabla_{\mathbf{y}} V^\star\|^2_{1/\gamma}}, \mu_2 \frac{\nabla_{\mathbf{z}} V^{\star\star}(\mathbf{p}_k)}{\|\nabla_{\mathbf{z}} V^{\star\star}\|^2_{1/\gamma}} \right), \quad (4)$$

where $\nabla_{\mathbf{x}} V(\mathbf{p}_k)$ designates the k^{th} entry of the vector $\nabla_{\mathbf{x}} V$ and analogously for y and z, and $\|a\|^2_{1/\gamma} = \sum_{k=0}^{N-1} \gamma_k a_k^2$.

While previous geometrical methods [RHC08, FTS08] precompute γ from skinning weights, making them hard to control and introducing artificial transitions between neighboring parts of the surface, we rather compute these weights on the fly using smooth unidimensional shape functions, oriented with respect to the articulated skeleton.

Another issue is to control not only the locality but also

the direction of the volume correction. Eq. (4) allows to compute a weighted correction following three orthogonal axes. If $\mu_i = 1/3$ was chosen, the correction would be isotropic which is generally not desired. Indeed, in the case of an inflating muscle the correction should follow some privileged direction. Having a local frame for each bone instead of a global one for the whole surface will add additional control. We therefore define an appropriate local frame at each joint.

Let us consider the hierarchical skeleton associated to the animation of the character. A convenient frame associated to the starting point of the bone b is denoted $(\mathbf{e}_0^b, \mathbf{e}_1^b, \mathbf{e}_2^b)$ and can be defined as follows: $\mathbf{e}_0^b$ is the unitary vector collinear to the direction of the parent bone. $\mathbf{e}_1^b$ given by $\mathbf{e}_0 \times \mathbf{d}$, where $\mathbf{d}$ is the unitary vector colinear to the current bone. At last, $\mathbf{e}_3^b = \mathbf{e}_1^b \times \mathbf{e}_2^b$ see also fig. 4. Note that this frame is undefined when two consecutives bones are colinear. This mostly occurs when the shape is not deformed, so no correction is needed for this bone. Ambiguous definitions of $\mathbf{e}_1^b$ and $\mathbf{e}_2^b$ can be avoided by projecting the respective directions from the previous time step to the plane orthogonal to $\mathbf{e}_0$.

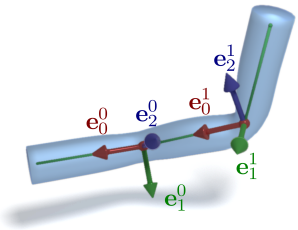

Figure 4: *Local frames defined for a articulated cylinder deformed by smooth skinning using three segments.*

Two set of parameters are now available at each joint to control intuitively the volume correction without the need of complex modeling tools:

1. The amount of correction (μ_0, μ_1, μ_2) (in percent) to apply along each axis $\mathbf{e}_0^b$, $\mathbf{e}_1^b$ and $\mathbf{e}_2^b$. This enables to favor specific orientations, in order to model anisotropic behavior.
2. The choice of the geometric distribution of the deformation defined by the weights γ. The knowledge of local frames enables to define γ locally. We will show in the subsection 3.3 that its distribution can easily be computed using composition of 1D profile curves intuitively defined by the user.

3.2. Exact iterative volume control

A deformation field $\mathbf{u}$ which locally preserves the volume around a given bone has been defined. However, the local change of volume ΔV induced by the motion of each skeleton frame b has to be known.

We use an iterative traversal of the skeleton hierarchy to compute $\Delta V(b)$ while decomposing the motion into a succession of individual joint rotations. Computing the exact volume change over the whole mesh iteratively enables to distribute the exact amount of correction locally without any restriction on overlaping influences.

The dependence of this deformation with respect to the hierarchy ordering is discussed later in section 5.

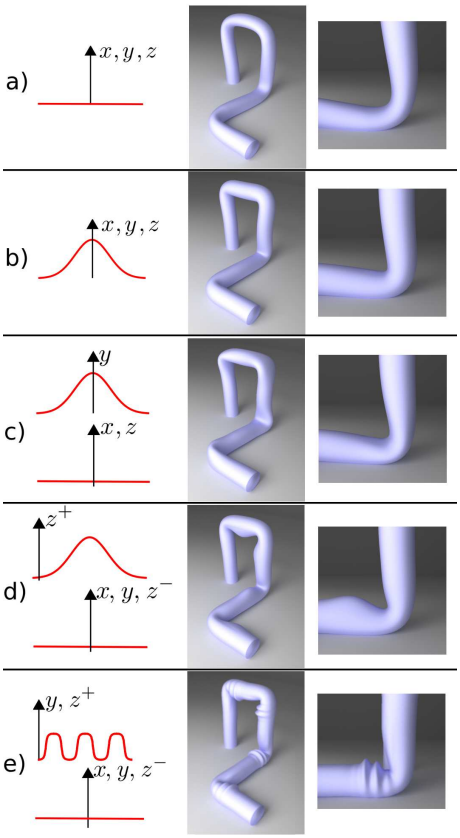

Figure 5: *Various profiles can be used to preserve volume while offering some local shape control. a) SSD reference. b) Isotropic inflation $(\mu_0, \mu_1\mu_2) = (1,1,1)/3$. c) Rubber effect $(\mu_0, \mu_1, \mu_2) = (0,1,0)$. d) Biceps-like muscle $(\mu_0, \mu_1\mu_2) = (0.1, 0.1, 0.8)$. e) Folds effect $(\mu_0, \mu_1\mu_2) = (0.0, 0.5, 0.5)$. The curves represents the γ distribution multiplied by the μ values.*

3.3. Specifying non-uniform volume corrections with 1D profile curves

In practice, the shape of the deformation induced by volume correction in each local frame (i.e. the set of weights) is controlled through smooth 1D profile curves, which can be independently defined for each of the three axes. To demonstrate how easy shaping this deformation is, let us explain

how to mimic the behavior of different materials on the example of a bent cylinder, see fig. 6. In the reference case, the cylinder is deformed using a classical SSD. In the following, $\mathbf{p}_k^L = (x_k^L, y_k^L, z_k^L)$ refers to coordinates of the vertex k in the current local frame $(\mathbf{e}_0^b, \mathbf{e}_1^b, \mathbf{e}_2^b)$. For each local frame a different set of parameters can be defined.

Local isotropic inflation

An isotropic inflation can be obtained using evenly distributed volume compensation by setting $\mu_0 = \mu_1 = \mu_2 = \frac{1}{3}$. The local effect of the inflation is set-up by the weight functions associated to the bones. In this case, a decreasing influence of the distance can be computed by $\gamma_k := \gamma_k(\mathbf{p}_k^L) = \exp(-(\|\mathbf{p}_k^L\|/\sigma)^2)$, where σ enables to vary smoothly between local to global deformation. Here σ was set to $1/4$ of the bone's length. This can be seen as the set of three Gaussian 1D curves, one for each local axis, see fig. 5b.

Local anisotropic inflation

A bent cylinder made of rubber typically exhibits an anisotropic deformation due to an extension oriented along the axis of the rotation.
This privileged axis is given by the vector $\mathbf{e}_1^b$. Setting $\mu_1 > \mu_0$ enables to apply this anisotropic effect while automatically adapting it to the orientation of the bend (note that this effect is further illustrated in the joined video, in the case of a rubber giraffe). $\mu_1 \in [0.5, 1]$ allows to distribute more or less deformation toward this direction. In fig. 5c the same γ function as the isotropic case is used. This time, only one Gaussian curve along the $\mathbf{e}_1^b$ direction is taken into account.

Unidirectional local anisotropic inflation

Mimicking the contraction of muscles such as biceps is of main interest for plausible animations of human-like characters. Muscles are generally attached along a single bone, so in the skeleton hierarchy, the deformation should only be defined along the parent bone's direction Here, the visible inflation is mainly acting in the positive direction of the axis orthogonal to the plane spanned by the rotation axis and the bone direction.

This axis is defined locally by $\mathbf{e}_2^b$. A larger value for μ_2 is sufficient to model this behavior, see fig. 5d. An example of such effect is provided by setting the weights to $\gamma_k := \gamma_k(\mathbf{p}_k^L) = \exp(-((\|\mathbf{p}_k^L - \mathbf{c}_y\|)/\sigma)^2)\max(z_k^L, 0)$. Here, the maximal deformation magnitude is centered along the father's bone by setting $\mathbf{c}_y = (0, c, 0)$ and σ still parameterizes how much the inflation propagates. The positive direction is specified through the sign of z_k^L. The parameterized curve is still a Gaussian one, but it is off-centered in local x^L coordinates and acts only for positive local z^L values.

Complex shapes

In practice, different effects can be achieved by using more complex profile curves or by combining them. For instance, fold effects which mimick cloth deformations can be achieved using oscillating curves. This can be obtained by setting $\gamma_k := \gamma_k(\mathbf{p}_k^L) = \sin^2(\omega\pi x_k^L)\exp(-(\|\mathbf{p}_k^L\|/\sigma)^2)$, where ω modulates the frequency of the oscillations, see fig. 5e.

4. Application to complex characters

Thereafter, various results of deformed animals will be presented by application of the exact volume preserving correction. Correction interactions and complex shape profiles will be applied on these animals to demonstrate the versatility of our method. Finally, a local refinement method is presented to seamlessly use complex deformation shapes on coarse meshes.

4.1. Automatic skinning weights computation

In complement to our volume control method, we propose an improved algorithm for automatically computing skinning weights, which can either be used with standard smooth skinning or with related methods such as dual quaternions.

Usually, skinning weights are painted by artists over the surface, but this is time-consuming and often un-intuitive due to their multiple dependencies to bones. In addition, their values are normalized to one. Badly defined weights lead necessarily to non realistic deformations. In order to limit artists work as much as possible, a good initialization of these values is highly appreciated. We propose an extension of an existing method to build correct skinning weights in a more robust manner, including the case of meshes of arbitrary convexity.

An automatic approach of skinning-weights computation was proposed by Baran *et al.* [BP07] and similarily by Weber *et al.* [WSLG07]. Herein the weigths are computed according the diffusion of the heat equation over the mesh in order to smooth out the influences of the frames. However the method needs an initialization of the weight before starting the diffusion smoothing. Weber *et al.* use user defined anchors, while Baran *et. al* suggest to initialize the values to $\frac{1}{d^2}$ when the segment linking the vertex to the bone lies entirely within the mesh volume, and zero otherwise. This automatical discontinuous definition works in most cases, but gives a bad initial value of the skinning weights if the mesh exhibits large non-convexities. Small variations in vertex positions may change dramatically the initial values when the line segment passes close to the mesh boundary. An example is shown in fig. 7.

We suggest to use instead a continuous geodesic volumetric distance as the initial guess to overcome these drawbacks. The shortest geodesic distance inside the volume between a vertex to a bone is computed using a precomputed voxelized model. With this initialization method, the weights' definition is intrinsically a continuous function of the character shape and enables a more robust computation. An example of complex shape badly initialized using Cartesian distance, but correctly processed by our method, is given in fig. 7.

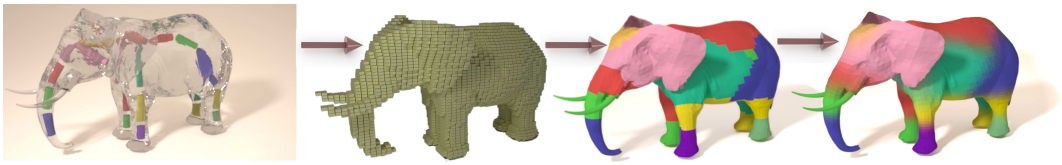

Figure 6: *Complete pipe-line of skinning weights computation. Left: Initial mesh + skeleton. Second picture: Voxelized domain bounded by the mesh. Third picture: Color encoded result after volumetric distance propagation (only the main influencing bone is shown). Right: Final result after surfacic diffusion.*

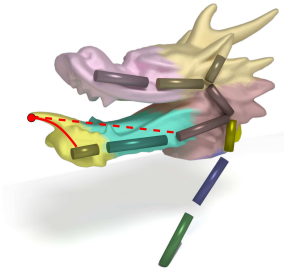

Figure 7: *Example where the Cartesian distance leads to an incorrect initialization. With this method, the highlighted vertex on the tongue would only depends on back throat bone (dashed straight line) leading to an associated skinning weight of one. The motion of this part of the tongue would therefore mainly be given by the orientation of the throat while the geodesic distance (curve line) associates it as expected with the jaw-bone.*

Note that the geodesic distance does not need to be highly accurate as long as the correct topological correspondences are roughly respected.

4.2. Application to animal animation

Let us illustrate the whole process by animating animal models designed by an artist. The extraction of the animated skeleton is performed using the [AHLD07] method. The skinning weights are computed fully automatically with the previous method. The volume is exactly corrected while merging corrections computed in different local frames. The latter are independently controlled thanks to our 1D profile curves.

A first result is presented for a giraffe toy. This model inspired from a real toy made of rubber-like material is modeled using the anisotropic inflation presented in section 3.3. Its neck is bent using only one rotation of a bone and the inflation direction is dynamically adapted to the orientation of this animated frame. The exact volume preservation gives visual results that are similar to the real shape we used (see fig. 1) and a purely geometric animation of this giraffe toy is provided to further illustrates this in the attached video.

An elephant model was also used to highlight other behaviors. First, a compression on the belly and a rotation of the front leg are applied. In the example depicted in fig. 9 the correction parameters were chosen using an unidirectional inflation along axis $\mathbf{e}_3^b$ with $\mu_1 = \mu_2 = 0.1$ and $\mu_3 = 0.8$. The influencing values were chosen such that the deformation of the shoulder propagates as far as the belly. The volume correction creates respective bulges due to these two motions that smoothly blend together.

We now use the same elephant model to simulate another complex deformation, namely the typical folds occurring on large flesh regions like a compressed belly. Such deformations can be seen as a combination of general inflation acting in one main direction with high frequency oscillations modeling the folding bulge. Our shape control parameters are set as follows: Let us denote γ^m the weights leading to a muscular effect using a unidirectional local anisotropic inflation profile. γ^f denote the weights leading to fold creation using an oscillating profile function. A blending of wrinkle appearance and inflation is finally achieved using $\gamma = (\alpha + (1-\alpha)\gamma^f)\gamma^m$.

Moreover, we can control the number of folds by taking into account the total size of the compression by setting $\alpha = dV/dV_{max}$. This leads finally to a linear blend between one single bulge when the change of the volume is small, and two bumps when the change of the volume is close to a maximum intensity, called dV_{max}. dV_{max} is known from the skinning deformation with maximal bend angle.

Using the weight functions of sect. 3.3 we therefore get the following γ function

$$\gamma_k = \left[\frac{dV}{dV_{max}} + (1 - \frac{dV}{dV_{max}})\sin^2(\omega\pi x_k^L)\right] e^{-(\mathbf{p}_k^L \cdot \sigma^\star)^2} \max(z_k^L, 0).$$

Finally $\sigma^\star = (\frac{1}{\sigma_0}, \frac{1}{\sigma_2}, \frac{1}{\sigma_3})$ enables to localize the deformation with an ellipse-like decreasing influence. Fig. 1 shows this deformation and a sequence of poses obtained by increasing bend angle parameters is shown in fig. 8 and in the attached video.

4.3. Adaptative refinement

Applying such deformation with high frequencies requires however a sufficient fine sampling of the mesh. It it there-

Figure 8: *Constant volume deformation of the belly, modeled as a two folds bulge. The cross-section views depicted at the bottom are cut along the left legs.*

fore needed to locally refine it. We only do this in the region of interest in order to keep a correction running at interactive rates. Moreover regions to subdivide should be automatically detected as they might change during the animation. We propose an adaptive subdivision algorithm associated to this volume preservation that localizes the region to subdivide by applying the algorithm twice.

In a first stage, the correction weights γ^b are computed with respect to every bones b. The maximal value is $\gamma_{max}(\mathbf{p}_k) = \max_b \gamma^b(\mathbf{p}_k)$. A segmentation is applied by choosing vertices k satisfying $\gamma_{max}(\mathbf{p}_k) >$ threshold. A classical subdivision scheme is then run over the region defined by the neighboring mesh faces of the eligible vertices. In a last stage, the complete volume preservation process is applied to the subdivided mesh.

In practice, during the animation, the highly deformed vertices are usually almost the same through time. Therefore the segmentation and subdivision steps do not need to be computed at each time step. An example of local refinement is shown in fig. 10.

4.4. Computational time

The different frame rates obtained when computing the animation sequences in the attached video are described in table 1. Note that the giraffe model only uses one compensation along the $\mathbf{e}_2^b$ direction, and is thus more efficient to compute than the elephant using the full three step correction.

5. Discussion and future work

A volume preserving method has been set-up using a three step solution which successively acts along the three axes of local frames. This iterative process permits to take into account the previous corrections while considering the next one, thus exactly solving the volume preservation problem.

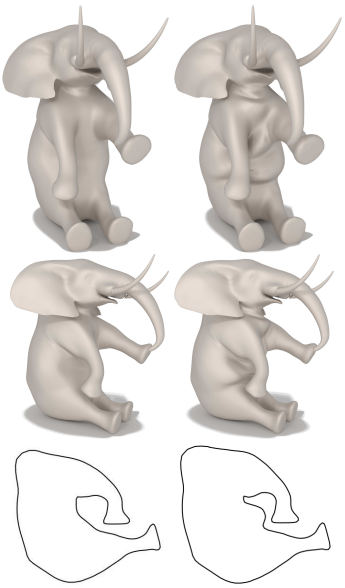

Figure 9: *Deformed elephant with overlapping volume correction influences. The bulge of the belly is blended to the one associated to the rotation of the leg. The reference SSD shape is shown on the left while the volume preserved mesh is on the right. The bottom part depicts the body's cross-section passing through the right leg.*

In order not to disturb the main correction step by previous small displacements, we suggest to process the correction along the axes with the largest value of μ first. However, the choice of the order is still arbitrary if several μ values are equal, and one could wonder how much this choice affects the final result.

To study the effect of changing the order in which the two axis directions are processed, we made a worse-case exper-

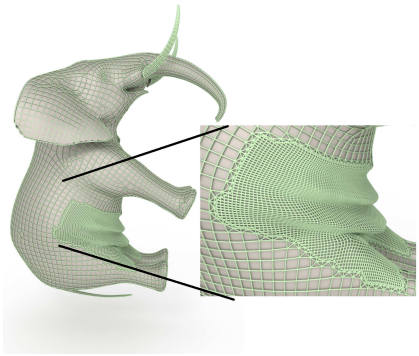

Figure 10: *Local subdivision of the mesh automatically defined in the region where bulges occur.*

	vertices	number of bones in rotation	computation cost
giraffe	1673	1	0.011s
elephant (1)	6646	1	0.053s
subdivided elephant (1)	13439	1	0.110s
elephant (2)	6646	17	0.407s

Table 1: *Elephant (1) designates the belly bulges effect, while elephant (2) corresponds to a full animation with multiple frames in rotations.*

iment where a cylinder bends of 90° with the deformation equally distributed along two axes : $\mu_1 = 0, \mu_2 = \mu_3 = \frac{1}{2}$. A first result is obtained using corrections along $\mathbf{e}_1^b$ and then $\mathbf{e}_2^b$, and a second one using the opposite order. The resulting deformations are depicted in cross-section in fig. 11a. As our results show, the meshes are indeed slightly different, with a displacement of vertices of at most 20% of the cylinder's radius. Although this difference is visually perceptible, both results remain plausible.

A second ordering choice is related to the fact that the iterative correction propagates throughout the skeleton hierarchy (see section 3.2). As the region in which a correction acts is un-bounded, a previous correction can influence the result of the current one. Therefore, the final correction vector $\mathbf{u}$ depends on the hierarchy of the skeleton and on the order in which the latter is traversed.

To quantify the effect of this order on the final shape, we conducted an experiment where a cylinder with three segments bends. The root bone is the middle one and the deformation chosen is similar to the previous test, but uni-axial with $(\mu_0, \mu_1, \mu_3) = (0, 0, 1)$. The volume correction profile is chosen such that the two corrections largely overlap each other. In the first case, the left bone is bent before the right one, while the opposite order is considered in the second case. The results, depicted in fig. 11b, show that the difference between the two cases is not visible. Numerically, the maximal magnitude of the displacement between the two meshes is less than 3% of the cylinder radius.

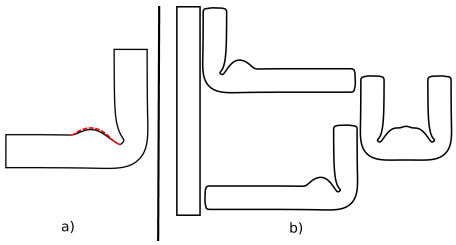

Figure 11: *a) Shapes obtained using a volume correction along $\mathbf{e}_1^b$ and then $\mathbf{e}_2^b$ (continuous black line) or in the opposite order (dashed red line). b) left: Original cylinder with three segments. Middle: First bone bent (two different choices). Right: Final shape after the second bending.*

In the current implementation, our results are not real time for large meshes animated with multiples bones inducing changes of volume (see previous section). However, calculation could be simplified if real-time performance is required

Firstly, the correction weights are currently computed on the fly with respect to every bone of the model using complex functions. In the majority of animations, the weights applied to the vertices are the same over time. Therefore, these functions do not have to be computed at every time step, and could even be pre-computed once if the main direction of correction does not vary over time.

A second process that can be optimized is the three step correction with respect to three orthogonal directions. If an approximate volume correction is sufficient, a first order Taylor approximation on the volume constraint developped in [SHB07] enables to express the solution of eq. 4 in only one step raising efficiency by a factor of three.

Third, a major flaw of the current method is the linear dependency of the computation time with the number of animated bones, due to the iterative approach and the fact that the actions of bones are considered un-bounded. However, in most cases of animated characters, only few bones have a visible effect on the character's volume. Therefore the computation could be performed only for these specific bones (such as one bone in the back in the elephant case, and four bones for the legs). Moreover, if few overlapping correction influences occur, the computation of the local difference of volume dV and its associated correction $\mathbf{u}$ can be computed independently in parallel and just summed up in the final stage. A single classical multi-threaded computer could therefore easily compute the correction of a few bones in an exact manner if there are no overlaps in the γ weights.

In the current implementation, the deformation applied

does not ensure that the final surface is fold-over free. Fast methods used for straightforward inflations around a bone [RHC08] cannot be used on complex deformation shapes (such as folds) where self collision can occur in a region influenced by the same articulation. In this case, a classical collision detection algorithm is required. When a self-intersection is detected, vertices can be moved following the direction $-\mathbf{u}$ until no intersections occur anymore in an iterative manner. Then a second step of correction would be performed, while fixing the positions of the vertices in contact by setting there corresponding γ values to zero.

Finally, a nice continuation of this work could be the addition of dynamic vibrations of fatty tissues. As explained in section 4.3, the flesh regions undergoing large deformations can be automatically extracted. The combination of this segmentation to dynamical deformations [LCA05, AS07] could then be used.

Lastly, interfacing this work with curve drawing tools would permit to easily and interactively sketch deformations over the surface.

6. Conclusion

An exact volume preserving method for animating skinned characters has been proposed. The final shape is intuitively controlled by distributing the amount of correction in meaningful local frames, thanks to 1-dimensional shape profiles. Complex bulge effects with multiple folds or anisotropic rubber-looking effects can be parameterized without any need for physical simulation. The region of influence of the applied corrections is not limited and can be tuned between extremely local deformations to global inflation, while still ensuring exact volume preservation. Associated to a robust automatic estimation of skinning weights, our approach perfectly fits into the standard animation pipeline, and reduces the artist's work load. The method has been presented as a way to keep the volume to a constant value, but could be used as well to animate changes of volume over time in order to model breathing effects for instance.

References

[AHLD07] AUJAY G., HETROY F., LAZARUS F., DEPRAZ C.: Harmonic skeleton for realistic character animation. In *Symposium on Computer Animatiom* (2007), pp. 151–160. 7

[Ale02] ALEXA M.: Linear combination of transformations. In *SIGGRAPH* (2002), pp. 380–387. 2

[AS07] ANGELIDIS A., SINGH K.: Kinodynamic skinning using volume preserving deformations. In *Symposium on Computer Animation* (2007). 2, 10

[BP07] BARAN I., POPOVIC J.: Automatic rigging and animation of 3d characters. *ACM Trans. Graph. 3*, 26 (2007). 6

[DMSB99] DESBRUN M., MEYER M., SCHRODER P., BARR A. H.: Implicit fairing of arbitrary meshes using diffusion and curvature flow. In *SIGGRAPH* (1999). 3

[Elb00] ELBER G.: *Linearizing the area and volume constraints*. Tech. rep., TECHNION Israel, 2000. 4

[FTS08] FUNCK W. V., THEISEL H., SEIDEL H.-P.: Volume-preserving mesh skinning. In *Workshop on Vision, Modeling and Visualization* (2008). 2, 4

[GOMP98] GONZALEZ-OCHOA C., MCCAMMON S., PETERS J.: Computing moments of objects enclosed by piecewise polynomial surfaces. *Transactions on Graphics 17*, 3 (1998), 143–157. 3

[HJCW06] HONG M., JUNG S., CHOI M.-H., WELCH S.: Fast volume preservation for a mass-spring system. *IEEE Comput. Graph. Appl. 26* (2006). 2, 4

[HML99] HIROTA G., MAHESHWARI R., LIN M. C.: Fast volume-preserving free form deformation using multi-level optimization. In *Symposium on Solid Modeling and Applications* (1999), pp. 234–245. 3

[HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L., TENG S., BAO H., GUO B., SHUM H.: Subspace gradient domain mesh deformation. In *SIGGRAPH* (2006), pp. 1126–1134. 2

[KCZO08] KAVAN L., COLLINS S., ZARA J., O'SULLIVAN C.: Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph. 27*, 4 (2008). 2

[KJP02] KRY P., JAMES D. L., PAI D. K.: EigenSkin: Real time large deformation character skinning in hardware. In *Symposium on Computer Animation* (2002), pp. 153–159. 2

[KZ05] KAVAN L., ZARA J.: Spherical blend skinning: a real-time deformation of articulated models. In *Symposium on Interactive 3D Graphics and games* (2005), pp. 9–16. 2

[LCA05] LARBOULETTE C., CANI M.-P., ARNALDI B.: Dynamic skinning: Adding real-time dynamic effects to an existing character animation. In *Spring Conference on Computer Graphics* (2005). 10

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose Space Deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH* (2000), pp. 165–172. 2

[MG03] MOHR A., GLEICHER M.: Building efficient, accurate character skins from examples. *Transactions on Graphics 22*, 3 (July 2003), 562–568. 2

[RHC08] ROHMER D., HAHMANN S., CANI M.-P.: Local volume preservation for skinned characters. *Comput. Graph. Forum 27*, 7 (2008). 2, 4, 10

[SHB07] SAUVAGE B., HAHMANN S., BONNEAU G.-P.: Volume preservation of multiresolution meshes. *Computer Graphics Forum 26*, 3 (2007), 275–283. 9

[SRC01] SLOAN P.-P., ROSE C., COHEN M.: Shape by example. In *Symposium on Interactive 3D graphics* (2001). 2

[WP02] WANG X. C., PHILLIPS C.: Multi-weight enveloping: Least-squares approximation techniques for skin animation. In *Symposium on Computer Animation* (2002), pp. 129–138. 2

[WPP07] WANG R. Y., PULLI K., POPOVIC J.: Real-time enveloping with rotational regression. *Transactions on Graphics 26*, 3 (2007). 2

[WSLG07] WEBER O., SORKINE O., LIPMAN Y., GOTSMAN C.: Context-aware skeletal shape deformation. *Comp. Graph. Forum 26* (2007). 6